

УДК 519.873

## The Synthesis of Spatial Filtering Algorithm with a Constant Level of Useful Signal

Valeriy N. Tyapkin\*

Dmitry D. Dmitriev†

Yuri L. Fateev‡

Nikolay S. Kremez§

Military Engineering Institute  
Siberian Federal University

Svobodny, 79, Krasnoyarsk, 660041

Received 10.02.2016, received in revised form 26.02.2016, accepted 22.03.2016

*The theoretical basis of two classical recursive algorithms is considered in the paper. They are beamforming algorithm based on the Kalman filter and the least squares algorithm based on QR decomposition. An algorithm that minimizes the noise at the output of an antenna array is proposed on the basis of these two algorithms. The algorithm allows one to maintain a constant level of the received useful signal.*

*Keywords: phased array, adaptive algorithm, Kalman filter, recursive least squares method, QR decomposition.*

DOI: 10.17516/1997-1397-2016-9-2-258-267.

## Introduction

Modern navigation receivers of global satellite systems have a significant drawback – low interference immunity [1]. To solve this problem, navigation receivers are based on phased array antennas that use adaptive beamforming techniques.

Pattern control is widely used in applications such as navigation, radar, wireless communication and others. The beamforming technique is one of the most popular methods to form digital antenna directional diagram. This technique allows one to form maximum in the antenna diagram in the direction of useful signal and to form null in the antenna diagram in the direction of interference. Then interference is suppressed and useful signal is accumulated from appropriate directions. The overall effect of this treatment is determined by the difference in useful signal and interference and statistical characteristics of useful signal and interference. Since beamforming takes place in digital form, it is necessary to use digital techniques that could be performed in real time. Therefore, the solution of this task depends on both hardware and software.

We consider theoretical foundations of digital beamforming algorithms such as the recursive Kalman filtering algorithm and the recursive least squares algorithm based on the QR decomposition (QR-RLS)[2].

On the basis of these two algorithms we propose an algorithm that minimizes the noise at the output of an antenna array and allows one to maintain a constant level of the desired signal.

\*tyapkin58@mail.ru

†dmitriev121074@mail.ru

‡fateev\_yury@inbox.ru

§Nkremez@sfu-kras.ru

## 1. Kalman filtering algorithm

The Kalman filtering algorithm is an important adaptive digital signal processing algorithm. It is an efficient recursive filter that estimates the state vector of a dynamic system and uses a series of incomplete and noisy measurements. The Kalman filter for recursive estimation of the state vector of a dynamic system is known a priori. In order to calculate the current state of a system it is necessary to know the current measurement and the previous state of the filter. Thus, the Kalman filter, like other recursive filters, operates in time rather than in the frequency domain. The Kalman filter produces not only estimates of the current state but also uncertainty (distribution density) of the state vector that is based on Bayes' formula of conditional probability.

Traditional Kalman filter has the following parameters:

$$\begin{aligned}\mathbf{K}(n) &= \mathbf{K}(n, n-1) - \mathbf{F}(n, n+1) \cdot \mathbf{G}(n) \cdot \mathbf{C}(n) \cdot \mathbf{K}(n, n-1), \\ \mathbf{G}(n) &= \mathbf{F}(n+1, n) \cdot \mathbf{K}(n, n-1) \cdot \mathbf{C}^H \cdot [\mathbf{C}(n) \cdot \mathbf{K}(n, n-1) \cdot \mathbf{C}^H(n) + \mathbf{Q}_2(n)]^{-1}, \\ \alpha(n) &= \mathbf{y}(n) - \mathbf{C}(n) \cdot \mathbf{x}(n|y_{n-1}),\end{aligned}\quad (1)$$

where  $\mathbf{C}(n)$  is the observation matrix,  $\mathbf{F}(n, n+1)$  is the state transition matrix.

In the case of not induced noise dynamic model the process converges to white noise with zero mean and unit variance. Therefore, some parameters of the state space converge to a constant and they are

$$\begin{aligned}\mathbf{F}(n+1, n) &= \sqrt{\lambda} \mathbf{I}, \quad \mathbf{C}(n) = \mathbf{u}^H(n), \quad \mathbf{Q}_2(n) = 1, \\ \mathbf{K}(n) &= \frac{\mathbf{K}(n-1)}{\lambda} - \frac{\mathbf{g}(n) \cdot \mathbf{u}(n)^H \cdot \mathbf{K}(n-1)}{\sqrt{\lambda}}, \\ \mathbf{g}(n) &= \frac{\mathbf{K}(n-1) \cdot \mathbf{u}(n)}{\sqrt{\lambda} (1 + \mathbf{u}(n)^H \cdot \mathbf{K}(n-1) \cdot \mathbf{u}(n))}, \\ \alpha(n) &= y(n) = \mathbf{u}^H(n) \cdot \mathbf{x}(n|y_{n-1}).\end{aligned}\quad (2)$$

Using expressions (2), we obtain parameters of the Kalman filter:

$$\begin{aligned}\mathbf{g}(n) &= \frac{\mathbf{K}^{-1}(n) \cdot \mathbf{u}(n)}{\sqrt{\lambda}}, \\ \mathbf{K}^{-1}(n) &= \lambda \cdot \mathbf{K}^{-1}(n-1) + \mathbf{u}(n) \cdot \mathbf{u}(n)^H, \\ \mathbf{K}^{-1}(n) \cdot \mathbf{x}(n+1|y_n) &= \frac{\mathbf{K}^{-1}(n-1) \cdot \mathbf{x}(n|y_{n-1})}{\sqrt{\lambda}} + \frac{\mathbf{u}(n) \cdot y(n)}{\sqrt{\lambda}}.\end{aligned}\quad (3)$$

Then, by rewriting the expression  $\mathbf{K}^{-1}(n) = \mathbf{K}^{-H/2}(n) \cdot \mathbf{K}^{-1/2}$  and using the lemma on matrix inversion, we define the Kalman filter in the form of pre-array and post-array as follows

$$\begin{bmatrix} \sqrt{\lambda} \cdot \mathbf{K}^{-H/2}(n-1) & \sqrt{\lambda} \cdot \mathbf{u}(n) \\ \mathbf{x}^H(n|y_{n-1}) \cdot \mathbf{K}^{-H/2}(n-1) & y^*(n) \\ \mathbf{0}^T & \mathbf{1} \end{bmatrix} \theta(n) = \begin{bmatrix} \mathbf{K}^{-H/2}(n) & \mathbf{0} \\ \mathbf{x}^H(n+1|y_n) \cdot \mathbf{K}^{-H/2}(n) & \frac{\alpha^*(n)}{\sqrt{r(n)}} \\ \sqrt{\lambda} \cdot \mathbf{u}^H(n) \cdot \mathbf{K}^{1/2}(n) & \sqrt{r(n)} \end{bmatrix}. \quad (4)$$

With the help of relation (4) coefficients of the Kalman filter can be recursively computed.

2. Recursive least squares algorithm based on the QR-decomposition (QR-RLS)

A recursive algorithm for adaptive filtering that uses QR-decomposition estimates the filter coefficients in the current step in terms of the filter coefficients calculated in the previous step. Due to its recursive nature the algorithm is called the recursive least squares algorithm (RLS). The basic idea of the QR-factorization is to convert a matrix into a product of an orthogonal matrix  $\mathbf{Q}$  and an upper triangular matrix  $\mathbf{R}$ .

Let us assume that  $m$  and  $n$  are any positive integers greater than 0. The QR-decomposition of matrix  $\mathbf{A}$  of size  $m \times n$  for every  $m > n$  has the form

$$\mathbf{A}=\mathbf{Q}\mathbf{R}. \tag{5}$$

Let  $\mathbf{Q}$  be a unitary matrix, i.e.,  $\mathbf{Q} \cdot \mathbf{Q}^H = \mathbf{I}$ , where  $\mathbf{I}$  is the identity matrix and  $\mathbf{R}$  is  $m \times n$  upper triangular matrix. Then equation (5) can be written in the divided form

$$\mathbf{A} = \mathbf{Q}\mathbf{R} = \mathbf{Q} \begin{bmatrix} \mathbf{R}_1 \\ 0 \end{bmatrix} = [\mathbf{Q}_1 \quad \mathbf{Q}_2] \begin{bmatrix} \mathbf{R}_1 \\ 0 \end{bmatrix} = \mathbf{Q}_1 \mathbf{R}_1, \quad (6)$$

where  $\mathbf{R}_1$  is an  $n \times n$  triangular matrix,  $\mathbf{Q}_1$  is an  $m \times n$  matrix and  $\mathbf{Q}_2$  is an  $m \times (m - n)$  matrix.

Matrix  $\mathbf{R}$  can be obtained with the use of Givens rotations [3]. A Givens rotation is represented by a transformation matrix  $\mathbf{G}_{ij}$  of the form

$$\mathbf{G}(i,j) = \begin{bmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 0 & & & & & \\ & 0 & & 1 & & & & 0 \\ & & c & & \dots & & s & \\ & & & 1 & & & & \\ & & \vdots & & \ddots & & \vdots & \\ & & & & & 1 & & \\ & & -s & & \dots & & c & \\ & & & & & & & 1 \\ & & & & & & & & \ddots & \\ & 0 & & & & & & & & & 1 \end{bmatrix} \quad (7)$$

where  $c$  and  $s$  appear at the intersections of  $i$ th and  $j$ th rows and columns. These elements satisfy the following condition

$$s^2 + c^2 = 1. \quad (8)$$

Matrix  $\mathbf{G}_i$  is an orthogonal matrix and it represents the plane rotation because elements  $c$  and  $s$  can be represented as  $c = \cos(\varphi)$  and  $s = \sin(\varphi)$ . Properly selecting elements  $c$  and  $s$  and multiplying matrix  $\mathbf{A}$  by transformation matrices  $\mathbf{G}_1, \mathbf{G}_2, \dots, \mathbf{G}_{n-1}$ , one can reduce matrix  $\mathbf{A}$  to an upper triangular matrix. Now we proceed to the QR-RLS algorithm.

Assume that the amplitude of navigation signal is significantly lower than the intensity of the interference. Then, the signal does not have significant influence on the estimate of the correlation matrix of the interference  $\bar{\Phi}$ .

Suppose that at the inputs of  $M$ -element phased array there is an interference with the instantaneous oscillations  $u_1(t), u_2(t), \dots, u_M(t)$ . Oscillation phases are shifted relative to each other due to the path difference  $\Delta D$  from element to element by an amount  $\Delta\varphi = \frac{2\pi}{\lambda} \cdot \Delta D$ . According to the Kotelnikov theorem, interfering signals at the output of the  $i$ -th element of a phased array can be represented as a set of discrete samples of instantaneous amplitude  $u_{il}$  with sampling period  $T_D = 1/2f_m$  ( $f_m$  is the maximum frequency in the spectrum of the received interference signal) (Fig. 1). Here,  $i = 1, 2, \dots, M$ ;  $l = 1, 2, \dots, k$  are the numbers of discrete samples.

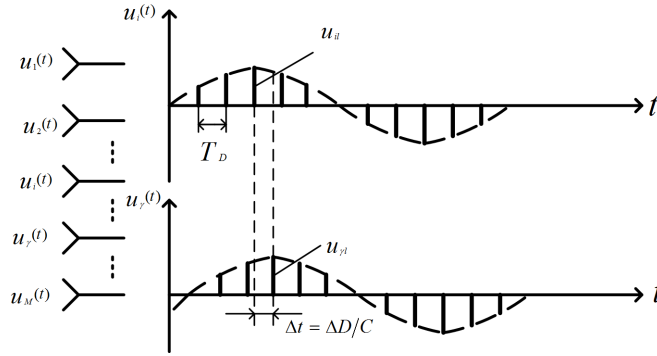


Fig. 1. Sampling signal interference

The cross-correlation between signals of the channels  $i$  and  $j$  is defined as the arithmetic mean of the correlation points  $u_i u_j$  over the number of counts  $l$  from 1 to  $k$  —  $\varphi_{ij} = \frac{1}{k} \sum_{l=1}^k u_{il} u_{jl}$ .

Upon introducing the complex amplitudes of instantaneous discrete values of  $\mathbf{u}_{il}$  and  $\mathbf{u}_{jl}$ , integrated mutual correlation time is defined as

$$\hat{\mathbf{V}} = \frac{1}{k} \sum_{l=1}^k u_{il} u_{jl}^* / 2. \quad (9)$$

The weight vector at stage  $n$  is defined as

$$\mathbf{w}(n) = [w_0(n), w_1(n), \dots, w_{M-1}(n)]^T. \quad (10)$$

The objective function for the RLS algorithm is

$$\delta(n) = \sum_{i=1}^n \lambda^{n-i} |e(i)|^2, \quad (11)$$

where  $e(i) = d(i) - \mathbf{w}^H(n) \cdot \mathbf{u}(i)$ . Let  $\mathbf{w}^H(n)$  be the optimum value such that the target function  $\delta(n)$  is minimal. This value satisfies the normal equation of the form [4]

$$\Phi(n) \cdot \mathbf{w}(n) = \mathbf{z}(n), \quad (12)$$

where  $\Phi(n) = \sum_{i=1}^n \lambda^{n-1} \cdot \mathbf{u}(i) \cdot \mathbf{u}(i)^H$ ,  $\mathbf{z}(n) = \sum_{i=1}^n \lambda^{n-1} \cdot \mathbf{u}(i) \cdot d^*(i)$ .

The last two equations can be written in a recursive form as follows

$$\begin{aligned}\Phi(n) &= \lambda \cdot \Phi(n-1) + \mathbf{u}(n) \cdot \mathbf{u}(n)^H, \\ \mathbf{z}(n) &= \lambda \cdot \mathbf{z}(n-1) + \mathbf{u}(n) \cdot \mathbf{d}^*(n).\end{aligned}\quad (13)$$

Next we use the lemma on matrix inversion. The inverse of  $\mathbf{A} + \alpha \mathbf{B} \mathbf{B}^{*T}$  has the form [5]

$$(\mathbf{A} + \alpha \mathbf{B} \mathbf{B}^{*T})^{-1} = \mathbf{A}^{-1} - \frac{\alpha \mathbf{A}^{-1} \mathbf{B} \mathbf{B}^{*T} \mathbf{A}^{-1}}{1 + \alpha \mathbf{B}^{*T} \mathbf{A}^{-1} \mathbf{B}}. \quad (14)$$

Using (14), we obtain

$$\mathbf{P}(n) = \frac{\mathbf{P}(n-1)}{\lambda} - \frac{\mathbf{k}(n) \cdot \mathbf{u}(n)^H \cdot \mathbf{P}(n-1)}{\lambda}, \quad (15)$$

$$\text{where } \mathbf{P}(n) = \Phi^{-1}(n), \quad \mathbf{k}(n) = \frac{\lambda^{-1} \cdot \mathbf{P}(n-1) \cdot \mathbf{u}(n)}{1 + \lambda^{-1} \cdot \mathbf{u}(n)^H \cdot \mathbf{P}(n-1) \cdot \mathbf{u}(n)} = \mathbf{P}(n) \cdot \mathbf{u}(n) = \Phi^{-1}(n) \cdot \mathbf{u}(n).$$

Using relations given above, we can obtain expression for the weight vector in the constraint equation (4):

$$\begin{aligned}\mathbf{w}(n) &= \Phi^{-1}(n) \cdot \mathbf{z}(n), \quad \mathbf{w}(n) = \mathbf{P}(n) \cdot \mathbf{z}(n), \\ \mathbf{w}(n) &= \lambda \cdot \mathbf{P}(n) \cdot \mathbf{z}(n-1) + \mathbf{P}(n) \cdot \mathbf{u}(n) \cdot \mathbf{d}^*(n), \\ \mathbf{w}(n) &= \mathbf{P}(n-1) \cdot \mathbf{z}(n-1) - \mathbf{k}(n) \cdot \mathbf{u}(n)^H \cdot \mathbf{P}(n-1) \cdot \mathbf{z}(n-1) + \mathbf{P}(n) \cdot \mathbf{u}(n) \cdot \mathbf{d}^*(n), \\ \mathbf{w}(n) &= \mathbf{w}(n-1) - \mathbf{k}(n) \cdot \mathbf{u}(n)^H \cdot \mathbf{w}(n-1) + \mathbf{P}(n) \cdot \mathbf{u}(n) \cdot \mathbf{d}^*(n), \\ \mathbf{w}(n) &= \mathbf{w}(n-1) + \mathbf{k}(n) \cdot \xi^*(n),\end{aligned}\quad (16)$$

where  $\xi^*(n) = d(n) - \mathbf{w}^H(n-1) \cdot \mathbf{u}(n)$ . From equations (15) and (3) one can observe correspondence between the RLS algorithm and the Kalman filter. Let us rewrite  $\Phi(n)$  as

$$\Phi(n) = \Phi^{-\frac{1}{2}}(n) \cdot \Phi^{-\frac{H}{2}}(n), \quad (17)$$

and introduce a new variable

$$\mathbf{p}(n) = \Phi^{\frac{H}{2}}(n) \cdot \mathbf{w}(n) = \Phi^{-\frac{1}{2}}(n) \cdot \mathbf{z}(n). \quad (18)$$

Then the RLS algorithm can be represented in the form of pre-array and post-array similar to the previously described Kalman algorithm:

$$\begin{bmatrix} \sqrt{\lambda} \cdot \Phi^{\frac{1}{2}}(n-1) & \mathbf{u}(n) \\ \sqrt{\lambda} \cdot \mathbf{p}^H(n-1) & d(n) \\ \mathbf{0}^T & 1 \end{bmatrix} \theta(n) = \begin{bmatrix} \Phi^{\frac{1}{2}}(n) & \mathbf{0} \\ \mathbf{p}^H(n) & \frac{\xi(n)}{\sqrt{\gamma(n)}} \\ \mathbf{u}^H(n) \cdot \Phi^{-\frac{H}{2}}(n) & \sqrt{\gamma(n)} \end{bmatrix}, \quad \mathbf{w}^H(n) = \mathbf{p}^H(n) \cdot \Phi^{-\frac{1}{2}}(n). \quad (19)$$

The QR-RLS algorithm is used to solve the problem of adaptive beamforming due to its recursive nature, computational efficiency and numerical stability [6].

### 3. An algorithm for minimizing the noise at the output of an antenna array and maintaining a constant level of the useful signal

One of the most promising methods for adaptation of the antenna arrays is the method of minimum variance distortionless response (MVDR). The MVDR algorithm allows one to increase

the signal to noise ratio in the desired direction while suppressing the interference of received signals.

The problem of minimizing the noise at the antenna array and maintaining a constant level of useful signal can be formulated as follows:

$$\min_{\mathbf{w}(n)} \sum_{i=1}^n \lambda^{n-i} |e(i)|^2, \quad (20)$$

where  $e(i) = \mathbf{w}^H(n) \cdot \mathbf{u}(i)$ , with the constrain

$$\mathbf{w}^H(n) \cdot \mathbf{s}(\theta_0) = 1, \quad (21)$$

where  $\mathbf{s}(\theta_0)$  is the vector that determines the direction of the source signal. Taking into account constraints (21), a weight vector for the antenna array can be written as follows [7]:

$$\mathbf{w}(n) = \frac{\Phi^{-1}(n) \cdot \mathbf{s}(\theta_0)}{\mathbf{s}^H(\theta_0) \cdot \Phi^{-1}(n) \cdot \mathbf{s}(\theta_0)}. \quad (22)$$

Let us define additional vector  $\alpha(n) = \Phi^{-\frac{1}{2}}(n) \cdot \mathbf{s}(\theta_0)$ . Then we rewrite expressions for the weight vector and error estimates as follows

$$\begin{aligned} \mathbf{w}(n) &= \frac{\Phi^{-\frac{H}{2}}(n) \cdot \alpha(n)}{\|\alpha(n)\|^2}, \\ \mathbf{e}(n) &= \frac{\alpha^H(n) \cdot \Phi^{-\frac{1}{2}}(n) \cdot \mathbf{u}(n)}{\|\alpha(n)\|^2}, \\ \mathbf{e}'(n) &= \alpha^H(n) \cdot \Phi^{-\frac{1}{2}}(n) \cdot \mathbf{u}(n). \end{aligned} \quad (23)$$

When parameters of the MVDR algorithm are set, it is possible to solve the problem of maintaining a constant level of useful signal in the QR-RLS algorithm and present it in the form of pre-array and post-array as follows

$$\begin{bmatrix} \sqrt{\lambda} \cdot \Phi^{\frac{1}{2}}(n-1) & \mathbf{u}(n) \\ \sqrt{\lambda} \cdot \alpha^H(n-1) & d(n) \\ \mathbf{0}^T & 1 \end{bmatrix} \theta(n) = \begin{bmatrix} \Phi^{\frac{1}{2}}(n) & \mathbf{0} \\ \alpha^H(n) & \frac{-e'(n)}{\sqrt{\gamma(n)}} \\ \mathbf{u}^H(n) \cdot \Phi^{-\frac{H}{2}}(n) & \sqrt{\gamma(n)} \end{bmatrix}. \quad (24)$$

After calculating parameters with the use of (24), one can estimate the error:

$$e(n) = \frac{-(-e'(n) \cdot \sqrt{\gamma(n)})}{\|\mathbf{a}(n)\|^2}. \quad (25)$$

## 4. Representation of the MVDR algorithm as a systolic array

Expressions obtained above show that the calculation of the response of a phased array antenna with the use of MVDR algorithm or its predecessor the QR-RLS algorithm involves matrix multiplication. This requires a series of Givens rotations. It is known that the consistent implementation of the matrix multiplication is inefficient. Real time signal processing systems that use the QR-RLS and MVDR beamforming algorithms require a more efficient matrix multiplication algorithm.

In 1978, Kung and Leiserson have proposed systolic arrays for matrix calculations in very large-scale integration systems. The systolic array is based on the method of triangular complex rotations and it significantly increases productivity of calculations in comparison with the traditional method of complex Givens rotations.

A systolic array system contains individual processing cells arranged in a specific structure. Each individual cell has its own function and local memory. Furthermore, only adjacent cells are connected to each other.

Data are written in a systolic array of processing cells on the entrance. Then data are processed, stored in local memory and transmitted to adjacent cells. This processing and sending the processed data in each cell lasts as long as the data stream reaches the end of the system where final results of calculation are obtained. The propagation of data through a systolic array resembles the pulse of the human circulatory system. The name is derived from the medical term systole as an analogy to the regular pumping of blood by the heart. This approach allows one to significantly reduce the time required to perform QR-decomposition using the same computational resources. The second advantage of the proposed scheme is that the QR-decomposition is executed such that the upper triangular matrix  $\mathbf{R}$  has only real diagonal elements. This greatly simplifies the subsequent inversion of the matrix  $\mathbf{R}$  with the use of back-substitution algorithm.

Consider the MVDR algorithm. After applying Givens rotations to (24), we obtain lower triangular matrix in the right hand side of (24) and set input vector  $\mathbf{u}(n)$  to zero. The number of elements in the input vector  $\mathbf{u}(n)$  corresponds to the number of antennas.

To set the input vector to zero, it is necessary to apply a number of Givens rotations. The number of rotations is the number of elements in the input vector because each Givens rotation sets one element of the input vector to zero. In general, the MVDR beamforming algorithm with  $K$  antennas requires  $K$  Givens rotations.

Calculations of Givens rotations can be performed in parallel because there is no data dependency between elements.

Let us consider expressions (7) and (8). One can see that each Givens rotation changes only certain elements of the pre-array, other elements are remained unchanged. To illustrate this we present the MVDR algorithm in the case of the antenna array consisting of three elements.

Step 1:

$$\begin{bmatrix} \Phi_{1,1} & 0 & 0 & u_1 \\ \Phi_{2,1} & \Phi_{2,2} & 0 & u_2 \\ \Phi_{3,1} & \Phi_{3,2} & \Phi_{3,3} & u_3 \\ \alpha_1 & \alpha_2 & \alpha_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos & 0 & 0 & -\sin' \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \sin & 0 & 0 & \cos \end{bmatrix}, \quad (26)$$

where

$$\begin{aligned} \Phi'_{1,1} &= (\Phi_{1,1} \cdot \cos) + (u_1 \cdot \sin), \\ u'_1 &= (\Phi_{1,1} \cdot (-\sin')) + (u_1 \cdot \cos) = 0, \\ \Phi'_{2,1} &= (\Phi_{2,1} \cdot \cos) + (u_2 \cdot \sin), \\ u'_2 &= (\Phi_{2,1} \cdot (-\sin')) + (u_2 \cdot \cos), \quad \Phi'_{2,2} = \Phi_{2,2}, \\ \Phi'_{3,1} &= (\Phi_{3,1} \cdot \cos) + (u_3 \cdot \sin), \\ u'_3 &= (\Phi_{3,1} \cdot (-\sin')) + (u_3 \cdot \cos), \\ \Phi'_{3,2} &= \Phi_{3,2}, \quad \Phi'_{3,3} = \Phi_{3,3}, \\ \alpha'_1 &= \alpha_1 \cdot \cos, \quad \alpha'_2 = \alpha_2, \\ \alpha'_3 &= \alpha_3, \quad \beta' = \alpha_1 \cdot (-\sin'). \end{aligned}$$

Step 2:

$$\begin{bmatrix} \Phi'_{1,1} & 0 & 0 & 0 \\ \Phi'_{2,1} & \Phi'_{2,2} & 0 & u'_2 \\ \Phi'_{3,1} & \Phi'_{3,2} & \Phi'_{3,3} & u'_3 \\ \alpha'_1 & \alpha'_2 & \alpha'_3 & \beta' \\ \beta' & \beta' & \beta' & \beta' \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos & 0 & -\sin' \\ 0 & 0 & 1 & 0 \\ 0 & \sin & 0 & \cos \end{bmatrix}, \quad (27)$$

where

$$\begin{aligned} \Phi''_{1,1} &= \Phi'_{1,1}, \quad u''_1 = 0, \\ \Phi''_{2,1} &= \Phi'_{2,1}, \quad \Phi''_{2,2} = (\Phi'_{2,2} \cdot \cos) + (u'_2 \cdot \sin), \\ u''_2 &= (\Phi'_{2,2} \cdot (-\sin')) + (u'_2 \cdot \cos) = 0, \\ \Phi''_{3,1} &= \Phi'_{3,1}, \quad \Phi''_{3,2} = (\Phi'_{3,2} \cdot \cos) + (u'_3 \cdot \sin), \\ u''_3 &= (\Phi'_{3,2} \cdot (-\sin')) + (u'_3 \cdot \cos), \\ \Phi''_{3,3} &= \Phi'_{3,3}, \quad \alpha''_1 = \alpha'_1, \quad \alpha''_2 = (\alpha'_2 \cdot \cos) + (\beta' \cdot \sin), \\ \alpha''_3 &= \alpha'_3, \quad \beta'' = ((\alpha'_2 \cdot (-\sin')) + (\beta' \cdot \cos)). \end{aligned}$$

Step 3:

$$\begin{bmatrix} \Phi''_{1,1} & 0 & 0 & 0 \\ \Phi''_{2,1} & \Phi''_{2,2} & 0 & 0 \\ \Phi''_{3,1} & \Phi''_{3,2} & \Phi''_{3,3} & u''_3 \\ \alpha''_1 & \alpha''_2 & \alpha''_3 & \beta'' \beta'' & \beta'' & \beta'' & \beta'' \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos & -\sin' \\ 0 & 0 & \sin & \cos \end{bmatrix}, \quad (28)$$

where

$$\begin{aligned} \Phi'''_{1,1} &= \Phi''_{1,1}, \quad u'''_1 = 0, \\ \Phi'''_{2,1} &= \Phi''_{2,1}, \quad u'''_2 = 0, \\ \Phi'''_{2,2} &= \Phi''_{2,2}, \quad \Phi'''_{3,1} = \Phi''_{3,1}, \quad \Phi'''_{3,2} = \Phi''_{3,2}, \\ \Phi'''_{3,3} &= (\Phi''_{3,3} \cdot \cos) + (u''_3 \cdot \sin), \\ u'''_3 &= (\Phi''_{3,3} \cdot (-\sin^*)) + (u''_3 \cdot \cos), \\ \alpha'''_1 &= \alpha''_1, \quad \alpha'''_2 = \alpha''_2, \\ \alpha'''_3 &= (\alpha''_3 \cdot \cos) + (\beta'' \cdot \sin), \\ \beta''' &= ((\alpha''_3 \cdot (-\sin^*)) + (\beta'' \cdot \cos)). \end{aligned}$$

$$\begin{bmatrix} \Phi'''_{1,1} & 0 & 0 & 0 \\ \Phi'''_{2,1} & \Phi'''_{2,2} & 0 & 0 \\ \Phi'''_{3,1} & \Phi'''_{3,2} & \Phi'''_{3,3} & 0 \\ \alpha'''_1 & \alpha'''_2 & \alpha'''_3 & \beta''' \\ \beta''' & \beta''' & \beta''' & \beta''' \end{bmatrix}. \quad (29)$$

Fig. 2 shows the block diagram of the triangular systolic array that may be used to calculate the QR-decomposition of 3-by-3 matrix.

During systolic cycle processors execute one operation and transfer data to adjacent nodes of the triangular systolic array. Fig. 2 also shows the input and output data streams. The input vector is loaded after every systolic cycle. The square in Fig. 2 denotes the inside cell. It is responsible for the application of the elementary Givens rotations to the relevant matrix elements in accordance with expression (7).

The error  $e(n)$  is calculated by the following formula:  $e(n) = \frac{-(-e'(n) \cdot \gamma^{-\frac{1}{2}}(n) \cdot \gamma^{\frac{1}{2}}(n))}{||a(n)||^2}$ .

The circle with the index "x" corresponds to this operation on the diagram shown in Fig. 2.

Other circles shown on the diagram denote border cells. These cells are responsible for the calculation of diagonal elements of the transformed matrix.



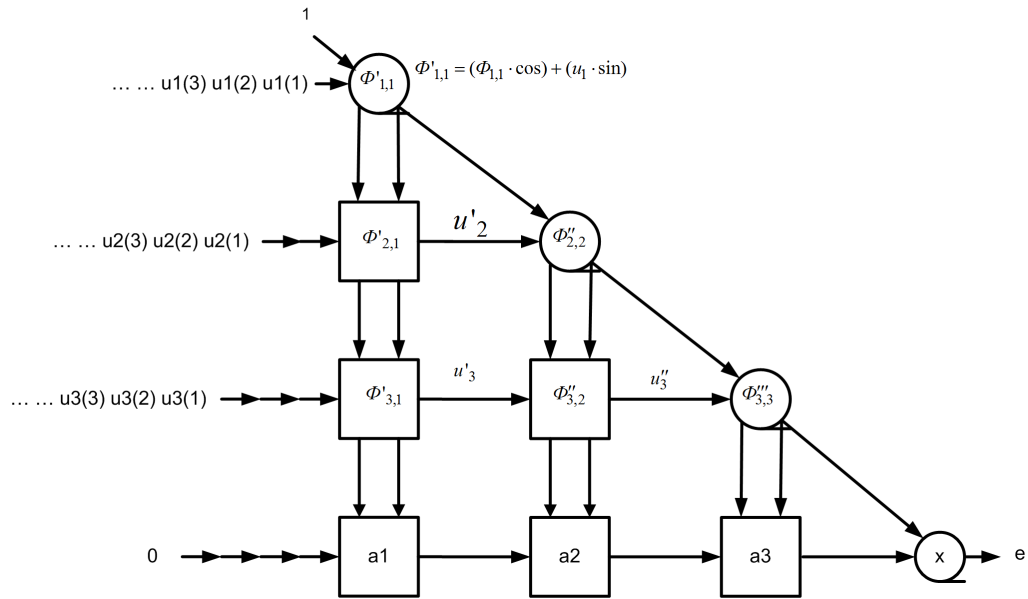


Fig. 2. Block diagram of the triangular systolic array

## Conclusion

The proposed structure of the triangular systolic array is optimized for implementation in large scale integrated circuits. It is based on triangular complex rotations and enables the efficient implementation of the QR-decomposition of complex matrices. The proposed structure can decrease the time of computation of QR-decomposition by 35% in comparison with the QR-RLS algorithm.

*This work was supported by the Ministry of Education and Science of the Russian Federation (agreement No. 14.575.21.0081, project RFMEFI57514X0081).*

## References

- [1] V.N.Tyapkin, D.D.Dmitriev, T.G.Moshkina, The potential noise immunity of consumer navigation equipment satellite navigation systems, *Vestnik SibGAU*, **43**(2012), no. 3, 113–119 (in Russian).
- [2] C.Dick, F.Harris, M.Pajic, D.Vuletic, Implementing a Real-Time Beamformer on an FPGA Platform: We designed a flexible QRD-based beamforming engine using Xilinx System Generator, *XCell Journal*, (2007), 36–40.
- [3] D.Bindel, J.Demmel, W.Kahan, On Computing Givens Rotations Reliably and Efficiently., *Acm Trans. Mathematical Software*, (2002), 206–238.
- [4] Ed. A.I.Perov, V.N.Kharisov., GLONASS. Principles of construction and operation. 4th ed., Revised, Moscow, Radio, 2010 (in Russian).
- [5] Ya.D.Shearman, Electronic systems: Basics of construction and theory. Resource Book. Ed. 2nd, Revised. and add., Moscow, Radio, 2007 (in Russian).

- [6] V.N.Tyapkin, D.D.Dmitriev, E.N.Garin, A.V.Sokolovsky, Management of amplitude-phase distributed adaptive phased array, *Industrial Automatic Control Systems and Controllers*, **4**(2013), 66–71 (in Russian).
- [7] V.N.Tyapkin, D.D.Dmitriev, V.G. Konnov, A.N.Fomin, The method of determining the vector of spectral coefficients of the likelihood ratio, *Vestnik SibGAU*, **43**(2012), no. 3, 76–79 (in Russian).

## **Синтез алгоритма пространственной фильтрации с сохранением постоянного уровня полезного сигнала**

**Валерий Н. Тяпкин**  
**Дмитрий Д. Дмитриев**  
**Юрий Л. Фатеев**  
**Николай С. Кремез**

Военно-инженерный институт  
Сибирский федеральный университет  
Академгородок, 13А, корпус 8, 660036

Россия

---

*В статье рассматриваются теоретические основы формирования диаграммы направленности на основе двух классических рекурсивных алгоритмов — алгоритм фильтра Калмана и алгоритм наименьших квадратов на основе QR-разложения. На основе этих двух алгоритмов синтезирован алгоритм минимизации шума на выходе антенной решетки, что позволяет поддерживать постоянный уровень принимаемого полезного сигнала.*

*Ключевые слова: фазированная антенная решетка, адаптивные алгоритмы, фильтр Калмана, рекурсивный метод наименьших квадратов, QR-разложение.*